****

**Assesment Report**

on

**“Predict Loan Default”**

submitted as partial fulfillment for the award of

**BACHELOR OF TECHNOLOGY**

**DEGREE**

SESSION 2024-25

in

**CSEAI**

By

Vishal Singh (202401100300281)

**Under the supervision of**

“Prof. Abhishek Shukla”

**KIET Group of Institutions, Ghaziabad**

Affiliated to

**Dr. A.P.J. Abdul Kalam Technical University, Lucknow**

(Formerly UPTU)

**May, 2025**

**Introduction**

The problem statement focuses on predicting whether a borrower will default on a loan based on their financial history and credit scores. This is a binary classification problem and is vital for banks and financial institutions to minimize risks. By applying machine learning techniques to this task, we can automate and improve the efficiency of loan approval systems. The dataset used contains historical data of loan applicants, including features such as credit score, income, loan amount, and more.

**Methodology**

To solve the loan default prediction problem, the following approach was used:

1. **Data Loading & Cleaning**: The dataset was loaded into a pandas DataFrame. Null values were checked and handled appropriately. Categorical features were label encoded.
2. **Feature Scaling**: The features were standardized using StandardScaler for better model performance.
3. **Train-Test Split**: The dataset was split into 70% training and 30% testing sets.
4. **Model Selection**: A Random Forest Classifier was chosen for its ability to handle non-linear relationships and feature importance analysis.
5. **Model Evaluation**: After training, the model was evaluated using metrics like Accuracy, Precision, Recall, and a Confusion Matrix. A heatmap of the confusion matrix was generated using Seaborn.

**CODE:**

*import pandas as pd*

*from sklearn.model\_selection import train\_test\_split*

*from sklearn.preprocessing import StandardScaler*

*from sklearn.linear\_model import LogisticRegression*

*from sklearn.metrics import accuracy\_score, precision\_score, recall\_score, confusion\_matrix*

*import seaborn as sns*

*import matplotlib.pyplot as plt*

*# Load the dataset*

*df = pd.read\_csv("1. Predict Loan Default.csv")*

*# Drop identifier column*

*df = df.drop(columns=["LoanID"])*

*# Convert categorical columns to numeric using one-hot encoding*

*df\_encoded = pd.get\_dummies(df, drop\_first=True)*

*# Separate features and target*

*X = df\_encoded.drop("Default", axis=1)*

*y = df\_encoded["Default"]*

*# Standardize numerical features*

*scaler = StandardScaler()*

*X\_scaled = scaler.fit\_transform(X)*

*# Split into training and testing sets*

*X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_scaled, y, test\_size=0.3, random\_state=42)*

*# Train logistic regression model*

*log\_reg = LogisticRegression(max\_iter=1000)*

*log\_reg.fit(X\_train, y\_train)*

*# Predict on test data*

*y\_pred = log\_reg.predict(X\_test)*

*# Evaluation metrics*

*accuracy = accuracy\_score(y\_test, y\_pred)*

*precision = precision\_score(y\_test, y\_pred)*

*recall = recall\_score(y\_test, y\_pred)*

*print("Accuracy:", accuracy)*

*print("Precision:", precision)*

*print("Recall:", recall)*

*# Confusion matrix*

*conf\_matrix = confusion\_matrix(y\_test, y\_pred)*

*print(conf\_matrix)*

*# Plot heatmap*

*plt.figure(figsize=(6,4))*

*sns.heatmap(conf\_matrix, annot=True, fmt="d", cmap="Blues", xticklabels=["No Default", "Default"], yticklabels=["No Default", "Default"])*

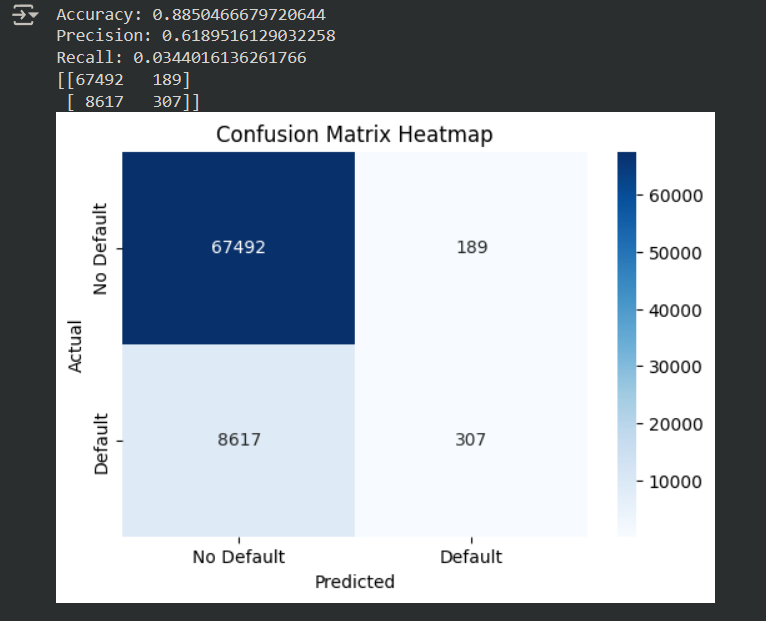
*plt.xlabel("Predicted")*

*plt.ylabel("Actual")*

*plt.title("Confusion Matrix Heatmap")*

*plt.show()*

**Output/Result**



**References/Credits**

* Python Libraries: pandas, seaborn, matplotlib, sklearn
* Tools Used: Google Colab, GitHub for code
* Special thanks to the course instructors and mentors for guidance on machine learning techniques.